IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 6, Issue 1, January 2026

Impact Factor: 7.67

PyqSpot: A Smart System for Engineering Preparation

Pisal Omkar Kisan[1], **Satalkar Vinay Balu**[2], **Pawar Avadhoot Subhash**[3],
hakare Samradnya Sanjay[4], **Prof. Kanade R. S.**[5]

Final Year Student, Department of Computer Engineering[1,2,3,4]

Professor, Department of Computer Engineering[5]

Hon Shri Babanrao Pachpute Vichardhara Trusts, GOI Faculty of Engineering Kashti, Ahmednagar, India

Abstract: *Engineering students at Savitribai Phule Pune University (SPPU) face significant challenges when preparing for exams because of the large syllabi and limited time. Research shows that 60-70% of the questions come from Previous Year Questions (PYQs), but students do not have access to a centralized collection of solved papers. Additionally, static PDF solutions do not support active learning or self-assessment. This paper introduces PYQSPOT, a new web system that provides a reliable archive of over 100 solved PYQs for various engineering branches at SPPU. The main technical contribution is the design and development of a Dynamic Mock Test Generator.*

This module analyses the static PDF solution archive, extracts individual question-answer pairs and creates a structured database. Using a Next.js and Express.js stack, the system automatically generates randomized multiple-choice mock tests based on subjects chosen by users, offering instant scoring and explanations. PYQSPOT effectively turns a static content archive into an interactive learning platform that offers a practical way to enhance student learning outcomes..

Keywords*:* PYQSPOT, E-Learning, Dynamic Mock Test, Previous Year Questions (PYQs), Automatic Question Generation, SPPU

I. INTRODUCTION

The academic environment for engineering students at Savitribai Phule Pune University (SPPU) has large syllabi and limited time. This situation leads to inefficient and high-stress "one-night" study patterns. A key finding is that 60-70% of exam questions are repeated from Previous Year Questions (PYQs), but students do not have access to a centralized collection of solved papers. To fill this gap, the PYQSPOT web platform was created. It includes over 100 verified PDF solutions for the Computer, Mechanical and AIDS engineering branches.

However, this static PDF model only encourages passive learning. This paper discusses the design and implementation of PYQSPOT's main technical feature: a Dynamic Mock Test Generator. This new module turns the static archive into an interactive, data-driven learning tool. It does this by parsing the PDFs, extracting question-answer pairs and automatically generating randomized mock tests based on specific subjects in MCQ format. This system offers a scalable and active self-assessment platform to improve student learning outcomes.

II. LITERATURE REVIEW

The current state of educational technology shows three main types of solutions, none of which fully meet the needs of SPPU students. First, large e-learning platforms like Chegg [3] and Course Hero [4] provide wide textbook solutions but do not have specialized, verified collections of SPPU-specific Previous Year Questions (PYQs). Second, academic research on Automatic Question Generation (AQG) [8] using Natural Language Processing (NLP) [9, 10] is often too complicated and is not part of a practical tool for students. Third, informal student-run blogs or shared drives might have PYQs, but this content is usually unverified, disorganized and does not include solutions or interactive features. This review shows a clear gap: no existing system combines a verified collection of solved, university-specific PYQs

DOI: 10.48175/IJARSCT-30729

with a practical Dynamic Mock Test Generator. PYQSPOT is meant to fill this gap by combining a reliable content base with a self-assessment tool.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The methodology for this project is divided into two phases. The first phase included designing, implementing and validating a Minimum Viable Product (MVP) to serve as the project's foundation. The second phase, which is the focus of this research paper, proposes designing an intelligent, active-learning module built on this proven foundation.

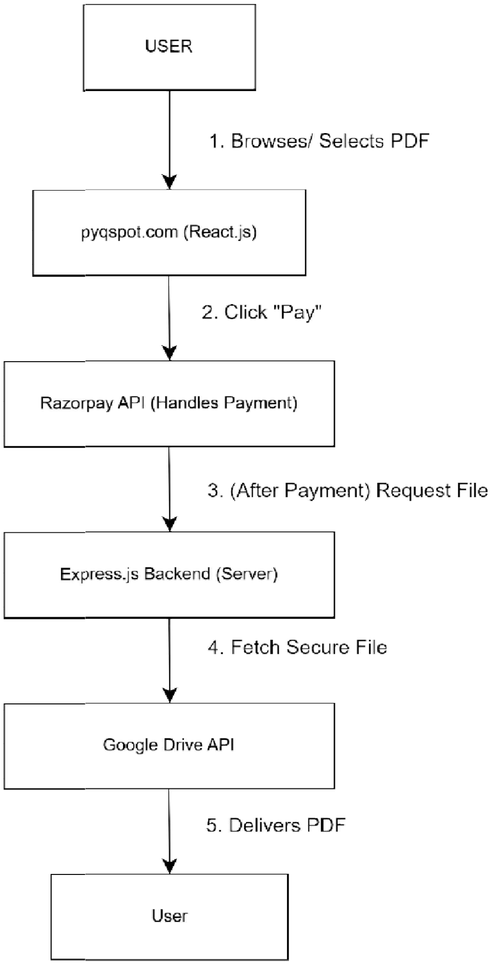


Fig. 1. Example of an unacceptable low-resolution image

The User (1) browses your React.js site. Then, (2) they click "Pay," and the site communicates with the Razorpay API to process the payment. After the payment, (3) the React site requests the file from your Express.js Backend. (4) Your backend securely contacts the Google Drive API to retrieve the PDF and (5) sends it to the user. This is a great and secure design.

1. Existing MVP Architecture (The Foundation)

The current operational platform, available at www.pyqspot.com, was created to test the main idea that students want and will pay for a centralized collection of solved PYQs. This MVP uses a modern web stack. A React.js frontend [11] offers a responsive user interface and an Express.js backend [13] server processes requests. The Razorpay Payment Gateway is integrated to handle transactions. The digital products, which include over 100 solved PDF solutions, are

DOI: 10.48175/IJARSCT-30729

stored on Google Drive. The Google Drive API automatically delivers the correct PDF to the user right after a successful payment. This setup has confirmed the market demand.

2. Proposed Architecture (The New Module)

To change the platform from a passive content repository, where users read PDFs, to an active learning tool that allows for self-assessment, we propose a new system. This new setup will include user authentication, a structured database and the Dynamic Mock Test Generator. We will develop this module using Next.js. This will improve the platform with server-side rendering and a stronger backend capability.

Content Ingestion and Database:

An Admin Panel will be created so administrators can upload new solved PDF solutions. After that, a backend Content Ingestion Pipeline will start. This script will use a parsing library to read the raw text from the PDFs. A custom algorithm will be designed to identify and extract individual question-and-answer pairs from the text using Regular Expressions (RegEx). These extracted pairs will be stored in a new MongoDB (NoSQL) database. Each question will be tagged with metadata such as branch, year and subject.

User Authentication:

A new user login and registration system will be set up, using NextAuth.js. This will let students create profiles, track their test history and monitor their performance over time.

Dynamic Mock Test Module:

This will be the main technical innovation for users. A logged-in student can choose a subject and year. The system will then search the MongoDB question bank, randomly select a set of questions (for example, 20) and present them in an interactive, timed, multiple-choice test. After finishing, the module will immediately grade the test, give a final score and display the correct answers. This creates the active-recall learning loop.

1. Structured Question Bank (The Input).

The main data structure is the Structured Question Bank (Q), which is a collection of all available questions taken from the static PDF archive. Each question $q_i \in Q$ is represented as a set of attributes. The Metadata is essential for filtering and is defined as:

$$q_i = (Test_i, Options_i, AnswerIndex_i, Metadata_i)$$

2. Question Selection Function.

The selection of questions for a specific mock test, T, involves two steps: Filtering and Randomization.

A. Filtering (Targeting):

The user selects a set of parameters for the target. $P_{target} = (Branch_{req}, Subject_{req}, Year_{req})$. The Filtered Question Set(Q_F) comes from Q with each question meeting the target criteria:

$$Q_F = \{\, q_i \in Q \mid Branch_i = \ Branch_{req} \ \wedge \ Subject_i = Subject_{req} \ \wedge \ Year_i = \ Year_{req} \,\}$$

B. Randomization (Test Generation)

A fixed number of questions, N_T, is taken from the filtered setQ_F. To create an effective study tool, the selection must be random. This helps avoid rote memorization and promotes active recall.

The Mock Test (T) is a part of Q_F with size N_T:

$$T = \left\{ q_1, q_2, \dots, q_{N_T} \right\} \subset Q_F$$

where the choice of each q_i is a random pick without replacement from Q_F.

3. Test Scoring Model (The Output)

The instant self-assessment feature needs a simple, reliable scoring system. Let A_U be the set of the student's submitted answers, where $a_{c,i}$ is the index of the option chosen for question q_i. Let A_C be the set of correct answers, where $a_{c,i}$ is the correct option index ($AnswerIndex_i$) from the Structured Question Bank.

DOI: 10.48175/IJARSCT-30729

The Performance Score (S) is calculated by comparing the student's answer for each question with the correct answer and giving points accordingly. The correct answers indicator function:

$$\delta_i = \begin{cases} 1\ if\ a_{u,i} = a_{c,i} \\ 0\ if\ a_{u,i} \neq a_{c,i} \end{cases}$$

If each correctly answered question receives 1 point, the total score (S) is the sum of correct answers:

$$S = \sum_{i=1}^{N_T} \delta_i$$

The final score shown to the student is expressed as a percentage $S_{\%}$, is:

$$S_{\%} = \left(\frac{S}{N_T}\right) \times 100$$

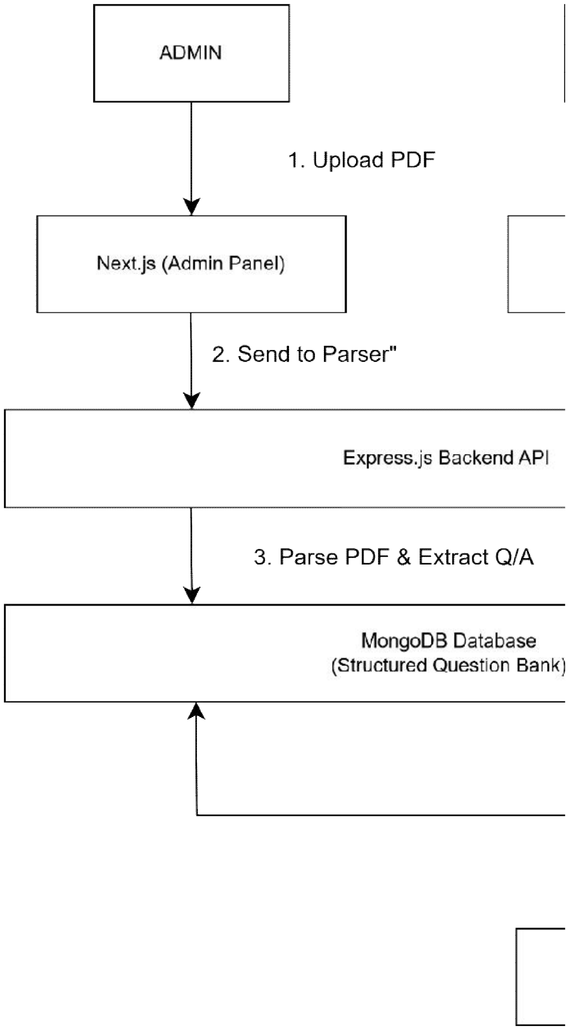


Fig. 2. Proposed System Architecture (The New Feature)

IV. IMPLEMENTATION AND RESULTS OF CURRENT MVP

The current, fully operational PYQSPOT system is the Minimum Viable Product (MVP) and the basis for this project. This system was created to confirm the main idea: that students would appreciate a centralized, paid collection of solved PYQs.

The MVP's structure uses a modern web stack. A React.js frontend offers a fast, responsive user interface, while an Express.js backend handles API requests. To manage payments, we integrated the Razorpay Payment Gateway API. This ensures secure and reliable transactions. Over 100 solved PDF solutions are stored safely on Google Drive, and we use the Google Drive API to automatically give access and start a download for users right after a successful payment. This setup does not need user sign-in, which simplifies the path from selection to download.

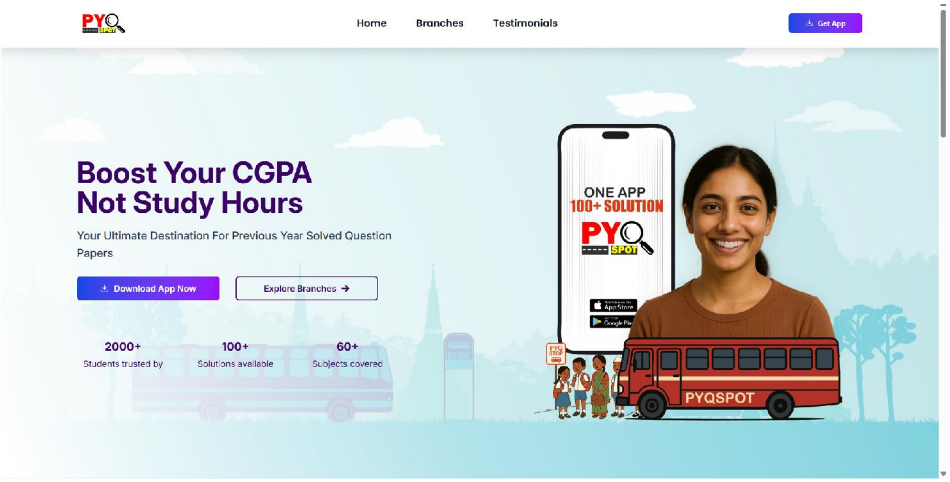
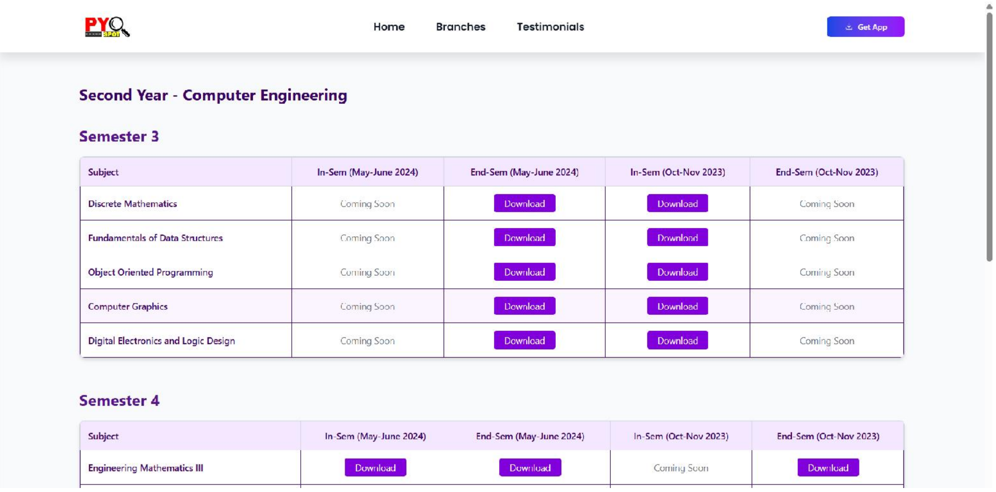


Fig. 3. A screenshot of UI Home page

The results of this setup have been very successful and strongly support the project's idea. The platform is now live at www.pyqspot.com and is actively used by students from the Computer, Mechanical, and AIDS engineering branches of SPPU. The system generates revenue, showing that students are willing to pay for high-quality, reliable, time-saving solutions. Positive feedback from students, who report getting good scores in less time, confirms the platform's value.



Second Year - Computer Engineering

Semester 3

Subject	In-Sem (May-June 2024)	End-Sem (May-June 2024)	In-Sem (Oct-Nov 2023)	End-Sem (Oct-Nov 2023)
Discrete Mathematics	Coming Soon	Download	Download	Coming Soon
Fundamentals of Data Structures	Coming Soon	Download	Download	Coming Soon
Object Oriented Programming	Coming Soon	Download	Download	Coming Soon
Computer Graphics	Coming Soon	Download	Download	Coming Soon
Digital Electronics and Logic Design	Coming Soon	Download	Download	Coming Soon

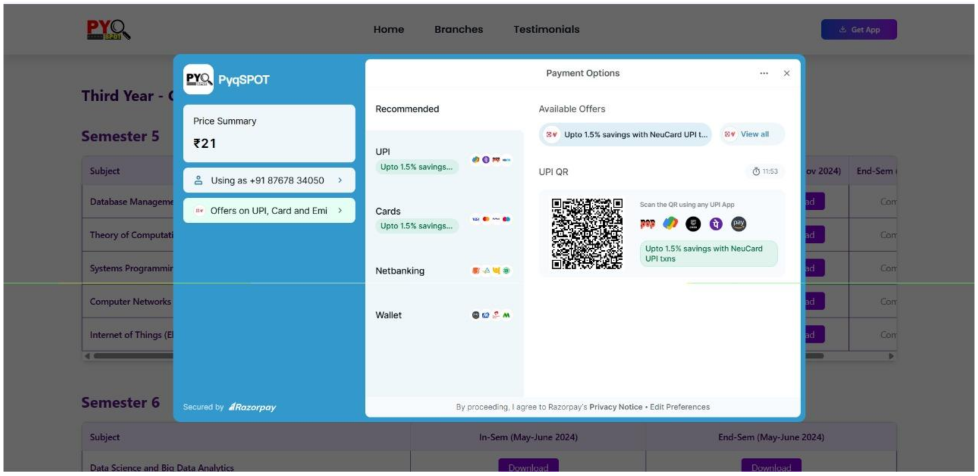
Semester 4

Subject	In-Sem (May-June 2024)	End-Sem (May-June 2024)	In-Sem (Oct-Nov 2023)	End-Sem (Oct-Nov 2023)
Engineering Mathematics III	Download	Download	Coming Soon	Download

Fig. 4. A screenshot of the subject/branch selection page

This proven success of the MVP establishes a user base and validates the main market. This justifies the next phase of the research: developing the Dynamic Mock Test Generator as described in the proposed architecture in Section 3. The current passive-learning model, which involves reading PDFs, will be improved with this new active-learning feature.

DOI: 10.48175/IJARSCT-30729



Price Summary
₹21

Using as +91 87678 34050

Offers on UPI, Card and Emi

Upto 1.5% savings with NeuCard UPI t...

Scan the QR using any UPI App

Upto 1.5% savings with NeuCard UPI txns

By proceeding, I agree to Razorpay's Privacy Notice • Edit Preferences

Fig. 5. A screenshot of the Razorpay payment popup

V. CONCLUSION AND FUTURE WORK

This paper presents a new two-phased approach to a major challenge in engineering education: the inefficiency of last-minute, passive studying. It explains the development and successful launch of the PYQSPOT web platform, a Minimum Viable Product that meets the clear need for a centralized and dependable source of solved Previous Year Questions (PYQs). The success of the current system, shown by its active user base and revenue generation, proves that students appreciate this time-saving resource.

Building on this solid foundation, the main contribution of this research is the formal design of a Dynamic Mock Test Generator. This proposed improvement will change PYQSPOT from a static content repository into a lively, interactive e-learning platform. By transforming the existing PDF archive into a structured question bank, the new module will give students a tool for active recall, self-assessment, and immediate feedback. This system directly tackles the "one-night study" problem by offering a tool that is both quick and effective.

Future Work

The next step is to complete the Dynamic Mock Test Generator. Once it is in place, there are many possibilities for this project. The long-term goal is to make PYQSPOT a single, central platform for all students in India.

Future work will concentrate on three main areas of growth. First, the platform will be expanded horizontally to include all other engineering branches at SPPU. After that, it will add courses like BCA, BBA, and MBA. Second, the platform will extend vertically to cover other universities in India, eventually reaching over 1,000 institutions nationwide.

Finally, we will look into more advanced features powered by AI. This includes using machine learning to study the frequency of past questions to predict likely topics for future exams. We will also develop personalized study plans that adjust to each student's performance on the mock tests.

VI. ACKNOWLEDGEMENT

We want to express our heartfelt thanks to our project guide, Prof. Kanade R.S. from the Department of Computer Engineering, for his valuable guidance, ongoing encouragement, and helpful feedback during this research. His support played a key role in the successful development of this project.

We also want to thank our institution, the Hon. Shri Babanrao Pachpute Vichardhara Trusts, GOI Faculty of Engineering Kashti, and the Department of Computer Engineering for providing the resources and academic environment needed to complete this work.

DOI: 10.48175/IJARSCT-30729

REFERENCES

[1]. Patel R.V., Sharma A.K., "E-Learning Platforms and Student Performance in Indian Technical Universities: A Comparative Study", International Journal of Engineering Education, 2021, 37 (2), 112-125.

[2]. Le, H.T., "Automatic Generation of Multiple-Choice Questions from Educational Texts Using Natural Language Processing", Journal of Educational Technology, 2020, 19 (3), 45-59.

[3]. Chegg, Inc., "Chegg Study: Expert Q&A and Textbook Solutions".

[4]. Course Hero, Inc., "Course Hero: 24/7 Homework Help".

[5]. Siddhartha S.S., Sreetama D., Mitrendu N.R., "Exploring the Efficiency of Aviation Companies in India using DEA Model", International Journal for Multidisciplinary Research (IJFMR), 2025, 7 (6), 1-7.

[6]. Singh, J., et al., "A Systematic Research of E-learning Platforms for Solving Challenges Faced by Indian Engineering Students", Journal of Engineering Education, 2021, 30(4), 582-595.

[7]. IBEF, "India's Edtech Surge: Growth in Online Education", India Brand Equity Foundation, 2024.

[8]. Mulla N., Gharpure P., "Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics and Applications", Progress in Artificial Intelligence, 2023.

[9]. Lan, Y., et al., "Survey of Natural Language Processing for Education: Taxonomy, Systematic Review and Future Trends", 2024. arXiv:2401.07518 [cs.CL].

[10]. Vercel, "Next.js by Vercel: The React Framework"

[11]. Adobe, "Adobe PDF Extract API Documentation", Adobe Developer.

[12]. Aslam, J., et al., "Navigating the E-Learning Platform with MongoDB, Express, React and Node", International Journal of Innovative Science and Research Technology, 2024, 9 (3), 1288-1290.

[13]. Ali, S. A. (2023). Desigining Secure And Robust E-Commerce Plaform For Public Cloud. Asian Bulletin of Big Data Management, 3 (1)

[14]. Altemimi, M. A. A., & Alasadi, A. H. H. (2022). Ecommerce based on Cloud Computing: The Art of State. European Journal of Information Technologies and Computer Science, 2 (5)

[15]. Lokhade, A. A., Gangadhar, S. D., Ingole, K., Kailashdevi, S. M. P. (2023). The Role of Digital Rights Management (DRM) in Modern Copyright Law. International Journal of Emerging Technologies and Innovative Research (IJETIR), 3 (11)